PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 13/38, 15/17

A1

(11) International Publication Number: WO 98/43178

(43) International Publication Date: 1 October 1998 (01.10.98)

(21) International Application Number: PCT/US98/05456

(22) International Filing Date: 20 March 1998 (20.03.98)

(30) Priority Data:

60/042,071	25 March 1997 (25.03.97)	US
09/000,711	30 December 1997 (30.12.97)	US

(71) Applicant: INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, P.O. Box 58119, Santa Clara, CA 95052-8119 (US).

(72) Inventors: KNAUERHASE, Robert, Conrad; 4926 S.W. Corbett Avenue #108, Portland, OR 97201 (US). TSO, Michael, Man-Hak; 5744 S.E. Preston Court, Hillsboro, OR 97123 (US).

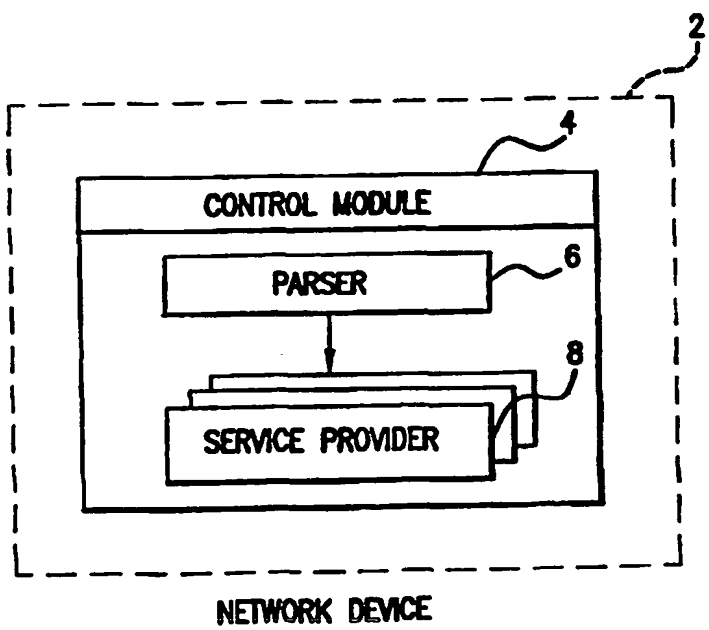
(74) Agents: HELLER, Paul, H. et al.; Kenyon & Kenyon, 1025 Connecticut Avenue, N.W., Washington, DC 20036 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published

With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SYSTEM FOR DYNAMICALLY CONTROLLING A NETWORK DEVICE



(57) Abstract

A system for dynamically controlling a network device is implemented in a network device (2) including a parser (6) and a service provider (8). The parser (6) includes instructions for selectively invoking the service provider (8) in response to a command parsed from an external input received by the network device (2).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

SYSTEM FOR DYNAMICALLY CONTROLLING A NETWORK DEVICE

This application claims the benefit of U.S. Provisional Application No. 60/042,071 titled "System for Dynamically Controlling a Network Proxy," filed March 25, 1997 by Robert C. Knauerhase et al. and assigned to Intel Corporation, the disclosure of which is expressly incorporated herein by reference.

Background of the Invention

Field of the Invention

The present invention relates generally to the field of computer networking, and in particular to a method and apparatus for enabling a device configured as a network proxy to perform a predetermined action in response to dynamic input.

Related Art

It is known to deploy a network proxy, or proxy server, as an intermediary between one or more client computers and an external network such as the Internet. Network proxies are described generally in Ian S. Graham, HTML Source Book: A Complete Guide to HTML 3.0 403 (2d ed. 1996). Most existing network proxies, however, lack any ability for command processing or dynamic configuration. Instead, network proxies are typically preconfigured using static control panels and settings stored in, for example, ".INI" files or MS-Windows registry entries. These settings remain unchanged for as long as the network proxy is running. Moreover, such network proxies lack any facility for dynamically receiving and acting upon commands or instructions that would effect a reconfiguration of the network proxy. This is because network proxies have typically been used only as a pipeline. For

example, a network proxy used for World-Wide Web (WWW) access simply passes HTTP requests received from a client computer to a server computer capable of servicing that request. Likewise, content which the network proxy receives from the server computer in response to such a request is simply passed to the requesting client computer. In other words, the network proxy does not inspect content passing through it.

In view of the foregoing limitations of existing technology, there is a need for a network proxy capable of acting upon information passed to it, whether it be a command embedded in a request originated by a client computer or content provided by a server computer. Although such dynamic control of network proxies is desirable, it is impractical to expect the expansive Internet infrastructure to quickly change to accommodate such a new capability. For this reason, it is desirable to implement this new capability in a way that does not require changes to existing client computers or server computers.

Summary of the Invention

The present invention relates to systems, methods and devices for dynamically controlling a network proxy. Embodiments may be used, for example, to dynamically effect network reconfigurations, to alter operational parameters of the network proxy, and to transcode network content prior to transmitting it to a client device.

According to one particular embodiment, a dynamically-controllable network device comprises a parser and a service provider. The parser includes instructions for selectively invoking the service provider in response to a command parsed from an external input received by the network device.

Brief Description of the Drawings

Fig. 1 is a schematic diagram illustrating a dynamically controllable network device according to an embodiment of the present invention.

Fig. 2 is a schematic diagram illustrating an embodiment of the present invention in which a dynamically controllable network device is implemented as a

network proxy.

Fig. 3 is a schematic diagram illustrating an embodiment of the present invention in which a dynamically controllable network device is implemented as a transcoding server.

Fig. 4 is a flow diagram illustrating a method for dynamically controlling a network device according to an embodiment of the present invention.

Detailed Description

According to a first embodiment of the present invention, illustrated schematically in **Fig. 1**, a means is provided by which a network device **2** (which may be any computer configured to act on behalf of another computer for network transactions) may be instructed to act in a particular manner based upon an externally-originating input. Such input may originate from, for example, a system administrator. Network device **2** is configured to inspect network traffic passing through it. In some instances, network device **2** may intercept commands and take certain predetermined responsive actions. In other instances, network device **2** may take certain actions based upon a characteristic of the content it receives, such as a datatype or MIME (Multipurpose Internet Mail Extensions) type. Such actions may include, for example, setting operational parameters on network device **2**, such as the size of a cache storage; remote management of network device **2**, such as retrieving/purging log files, obtaining system status information, and restarting network device **2**; and promulgating software upgrades.

In this embodiment, network device **2** includes a control module **4** having a parser **6** and a plurality of service providers **8**. Control module **4** may be implemented, for example, as a software module installed in network device **2**. Parser **6** is configured to act upon an external input received by control module **4**, such as a request for a network object generated by a client device or a reply to such a request provided by a network server device. In this particular embodiment, parser **6** is responsible for selectively invoking one or more of service providers **8** based upon a predetermined selection criterion. The predetermined selection criterion may

be, for example, a command embedded within a request, a characteristic of a received request or data object, a condition of network device **2** itself, combinations of the foregoing, and so on.

According to another embodiment of the present invention, illustrated schematically in **Fig. 2**, a dynamically-controllable network device may be implemented as a network proxy. A network proxy, or proxy server, is typically used in conjunction with so-called "firewall" software to protect a LAN (Local Area Network) from unauthorized access over the Internet. A firewall, typically installed on a gateway computer that links a LAN to the external world, restricts externally-originated network packets from entering the local network, thereby protecting the LAN from hazards such as unauthorized access. The firewall, however, also prevents network users from directly accessing external resources such as the Web. Network proxies are often used to address this shortcoming. See Graham, at 403.

Network proxies are usually configured to have free access to both internal LAN resources and external resources, and can safely pass data back and forth across the firewall. Users may then be given safe, though indirect, access to Web resources by configuring the user's Web browser to reference the network proxy instead of external target servers. When the Web browser is used to retrieve information from outside the firewall it sends a request to the network proxy, which then completes the request and returns the result to the requester. Id.

Referring now to **Fig. 2**, according to this embodiment a dynamically-controllable network proxy **34** manages the transfer of data from the Internet **18** to a network client **12**. Network client **12** and network proxy **34** may be any computers having suitable data communications and processing capabilities and capacities. Network client **12** communicates requests for information to, and receives information from, network proxy **34** over a client/server communications link **14**. Client/server communications link **14** may comprise, for example, a so-called "slow network" using, for example, POTS (Plain Old Telephone Service) dial-up technology or wireless connections. Alternatively, client/server communications link **14** may comprise a so-called "fast network," such as a LAN or WAN (Wide Area

Network), which is capable of operating at much higher speeds (for example, 5x to 10x) than are possible with slow networks. Combinations of these access methods are also possible. For example, network client **12** may use a POTS or wireless dial-up connection to a modem bank maintained by an ISP (Internet Service Provider), which is in turn connected to network proxy **34** over a LAN. Network proxy **34** communicates with computers resident on Internet **18** through server/network communications link **16**, which may comprise any suitable communications medium known in the art. Server/network communications link **16** is typically a much faster connection than client/server communications link **14**.

Network proxy **34** may be implemented, for example, as part of a network server, as a stand-alone computer in communication with a network server, or even as a distributed system of computers. Network proxy **34** may be coupled, for example, to a network server (not shown), an ISP's network, a corporate network, or anywhere on Internet **18**, and provides multiple users with a means to obtain content resident on Internet **18**. Network proxy **34** may also be coupled to one or more additional network proxies (not shown). Network proxy **34** differs significantly from known network proxies, which generally are little more than a conduit for requests to, and replies from, external Internet resources. Here, network proxy **34** not only examines such requests and replies, but may act on commands in the requests by, for example, dynamically reconfiguring one or more of its operational parameters.

Referring now to **Fig. 3**, according to yet another embodiment of the present invention, the dynamically-controllable network device may be implemented as a transcoding server **34** including a transcoder **20** includes a parser **22** and a plurality of transcode service providers **24**. Parser **22** is configured to act upon data received by transcoder **20**, such as a request for a network object generated by a client device or a reply to such a request provided by a content server device. In this particular embodiment, parser **22** is responsible for selectively invoking one or more of transcode service providers **24** based upon a predetermined selection criterion. As illustrated, transcoding server **34** may include an HTTP (HyperText Transfer

Protocol) remote proxy **36** capable of accessing Internet **18** over server/network communications link **16**. Using transcoder **20**, HTTP remote proxy **36** is capable of transcoding (for example, adding, changing and/or deleting) content received from Internet **18** prior to returning it to a requesting network client **12**.

Looking more closely at the embodiment in **Fig. 3**, parser **22** manages the transcoding of data to be transmitted from transcoding server **34** to network client **12**. To this end, parser **22** controls transcode service providers **24** to selectively transcode content based on a predetermined selection criterion. For example, one or more transcode service providers **24** may provide the capability to compress and/or scale different types of data content, such as image, video, or HTML (HyperText Markup Language). As is explained further below, one or more transcode service providers **24** may also be provided to act upon a command or other information contained in a request or a data object received by transcoding server **34**, or any other information determinable by transcoding server **34**, to effect the dynamic control functionality described herein.

As shown in **Fig. 3**, transcoding server **34** may also include a server-side cache memory **30** managed by a server-side cache interface **28**. Server-side cache memory **30** may be used to store both original and transcoded versions of content for later transmission to network client **12** without the need to re-retrieve the content from Internet **18** or to re-transcode the content.

Transcoding server **34** is coupled to network client **12** by client/server communications link **14**. Network client **12** includes a browser **32**, such as the Netscape Navigator v.3.0 browser (although the invention is not limited in this respect), which manages the presentation of data to a user. In this embodiment, network client **12** is "non-enabled," meaning no specialized transcoding software is preloaded on network client **12**.

Parser **22** may comprise a relatively simple, uniform interface to HTTP remote proxy **36**, and may provide an API (Application Programming Interface) for dynamically controlling transcoding server **34** and/or for transcoding data received by HTTP remote proxy **36**. Parser **22** manages one or more transcode service providers

24 that are accessed through a common SPI (Service Provider Interface). In this particular embodiment, parser 22 is designed in compliance with the Windows Open Systems Architecture (WOSA), and may be implemented as a Win32 DLL (Dynamic Link Library). The WOSA architecture, described in Readings on Microsoft Windows and WOSA (Microsoft Corp. 1995), enables additional transcode service providers 24 to be dynamically added to the system to provide new features, such as new or improved transcoding algorithms, while at the same time not requiring changing or retesting other software components in the system. This feature is especially beneficial where transcoding server 34 also interacts with "enabled" network clients equipped with specialized transcoding software. Transcoding server 34 may advantageously be configured flexibly enough to readily interact with both non-enabled and enabled network clients.

Like parser 22, server-side cache interface 28 may be modeled after a standard Get/Set interface. Server-side cache memory 30 essentially "owns" all cached objects, in that it manages the properties and storage of the objects and may invalidate any non-locked object at any time; however, the actual format of any given cached object is known only by parser 22 and its associated transcode service providers 24. Thus, for data integrity and transcoding efficiency purposes, all access to server-side cache memory 30 in this embodiment is through parser 22.

In this particular embodiment, parser 22 includes the following calls:

```
GetObject(URL, InParams, &OutParams, &OutStream, ...);
GetScaledObject(URL, InParams, &OutParams, &OutStream, Stage,
...);
PutObject(URL, InParamStruct, &InStream, &OutParams,
&OutStream, ...).
```

Parser 22 uses these calls to manage the provision of requested content to network client 12.

The GetObject() call is used to service non-enabled client requests, and returns a non-transcoded (i.e., original) version of a specified hypertext object. In this embodiment, transcoding server 34 assumes that each HTTP request has a unique thread that may be blocked until the request is satisfied. Accordingly, the

GetObject() call will block until it either returns the requested data stream or indicates failure with a cause (e.g., object does not exist). This ability to return a so-called standard hypertext object is advantageous for compatibility reasons, enabling embodiments of the present invention to be used with existing browsers that do not include support for certain transcoding functionality (e.g., advanced data compression), and enabling users to selectively retrieve non-transcoded versions.

The GetScaledObject() call is similar to GetObject(), and is also used to request an object from server-side cache memory **30**; however, it adds support for requesting a particular version of that object, such as a high-quality rendition. Unlike traditional caching proxies, transcode service providers **24** can use server-side cache memory **30** to store several different versions of an object to support clients with different communications and/or presentation capabilities. Thus, an additional "Stage" parameter may be used to indicate which version of the cached object is to be returned to network client **12**. Where transcode service provider **24** is configured to scale network content, it may use this parameter to request a version of a cached object having, for example, a default scaled quality, a refinement to a better-quality version, or the original non-scaled version.

In this embodiment, when network client **12** requests a hypertext object, HTTP remote proxy **36** uses either the GetObject() or GetScaledObject() call (depending on if network client **12** is capable of receiving scaled/transcoded datatypes) to retrieve the hypertext object from parser **22**. If the hypertext object is not found, parser **22** uses the CreateEntry() call to create an entry (in effect, a placeholder) in server-side cache memory **30** for the new object. The new entry is returned to HTTP remote proxy **36**, which requests the hypertext object from Internet **18**. As a data stream for the hypertext object is returned, HTTP remote proxy **36** calls parser **22** using the PutObject() call, passing into this call the new entry and the handle to the data stream to be placed into the entry. Parser **22** selects an appropriate transcode service provider **24** based, for example, on the content type of the data stream. In this context, the term content type encompasses a datatype, an HTTP MIME (Multipurpose Internet Mail Extensions) type, a content format, and so

on. The selected transcode service provider **24** uses a separate thread to read the incoming data stream, transcode it, and place it within the entry of server-side cache memory **30**. The current thread immediately returns to HTTP remote proxy **36**, which once again calls GetScaledObject() (or GetObject()). This case will always result in a cache hit. This thread then works simultaneously with the separate thread in the PutObject() to tunnel data (either original or transcoded) from transcoding server **34** to network client **12**.

Multiple-thread processing may be used to improve the efficiency of transcoding server **34** by not waiting for a hypertext object to be received in its entirety by HTTP remote proxy **36**, or added in its entirety to server-side cache memory **30**, before beginning to send the object to network client **12**. Another benefit of multiple-thread processing is that parser **22** may efficiently process requests for the same hypertext object from multiple network clients **12**. The hypertext object need only be retrieved from Internet **18** once, and appropriate versions may be transmitted to such multiple network clients **12** concurrently. It should be noted, however, that embodiments of the present invention may be implemented without multiple-thread processing.

In accordance with the embodiment illustrated in **Fig. 3**, transcoding server **34** may be dynamically controlled to effect any of a wide variety of operations, such as reconfiguring operational parameters of transcoding server **34**. For example, in order to reconfigure which network devices are permitted to access transcoding server **34**, an HTTP message may be sent to parser **22**, causing parser **22** to update (either itself or through transcoding service provider **24**) a security mechanism maintained by transcoding server **34** (for example, a table of valid network addresses). Similarly, an HTTP message could be sent to transcoding server **34** to effect a reconfiguration of one or more parameters used by transcoding service providers **24** to transcode content prior to transmission to network client **12**, such as instructing transcoding service provider **24** to delete all occurrences of a particular word from content passed through it. Other examples of the type of dynamic control that may be exerted using embodiments of the present invention include causing

transcoding server **34** to use more (or less) CPU resources for particular operations, reconfiguring cache memory **30**, and clearing cache memory **30**.

To further describe the operation of a dynamically-controllable network device, **Fig. 4** provides a flow diagram describing a general method for dynamically controlling network device according to an embodiment of the present invention. Processing begins with receipt by the dynamically-controllable network device of an input from another network device (**Step 20**). Where the dynamically-controllable network device comprises a network proxy, such other network device might comprise a client device or a content server. The dynamically-controllable network device then parses the received input in an effort to extract a command (**Step 30**). Assuming a command is found, the dynamically-controllable network device then performs some predetermined action in response to the command (**Step 40**). The method of this embodiment may be used, for example, to provide the dynamic control functionality described above.

A benefit of a dynamically-controllable network device according to embodiments of the present invention is that no changes need be made to existing network infrastructure. In an embodiment where the network device comprises a network proxy used for Web transactions, for example, commands may be sent to the network proxy via standard HTTP "POST" methods to special URLs. Accordingly, the only client software required to issue proxy commands is a standard Web browser, with no special configuration required of the client device other than specifying an HTTP proxy as a destination device in the manner currently known in the art. On the other hand, the network proxy can also accept commands from specialized client software if present. Similarly, the dynamically-controllable network proxy may freely interact with existing server devices. The proxy may generate its own forms and status replies for the client as needed, without the need for any specially-configured HTTP server software or CGI (Common Gateway Interface) script-processing capability. For example, the proxy can generate an HTML form that is displayed by the client's browser. The user may then fill in the form and submit it. The browser takes the filled-in form and uses the input to create a POST

communicating the results from the HTML form. The proxy may then act on the POST, possibly generating its own status replies so that the user is informed whether or not the input command(s) worked. Such status replies would then be displayed by the browser. Using an arrangement of this type, existing servers and clients may interact with the dynamically-controllable network proxy the same as they would with any other network proxy. Embodiments of the present invention may therefore be transparently implemented in existing network infrastructures, with no need for special communications ports or special protocols.

Embodiments of the present invention may be distributed, for example, as a set of instructions residing on a storage medium. Such a storage medium might be a memory of a computer; a piece of firmware; a portable storage device, such as a diskette or other magnetic storage device, or a CD-ROM; or any other medium on which it is known to store executable instructions.

Although the present invention has been described with reference to embodiments for accessing data from the Internet, persons skilled in the art will recognize that it is equally applicable to other networking environments. For example, embodiments of the present invention may be used to enhance data communications between a network client computer and an "intranet." An intranet typically is a secure corporate network modeled after the Internet architecture, and generally includes mechanisms for communicating with external networks such as the Internet.

The foregoing is a detailed description of particular embodiments of the present invention. The invention embraces all alternatives, modifications and variations that fall within the letter and spirit of the claims, as well as all equivalents of the claimed subject matter. For example, persons skilled in the art will readily recognize that the functionality described herein may be implemented in virtually any network device capable of receiving HTTP messages, including content servers and client devices. Moreover, embodiments of the present invention may be applied to communications protocols other than HTTP. Persons skilled in the art will recognize

from the foregoing detailed description that many other alternatives, modifications and variations are possible.

What Is Claimed Is:

1. A dynamically-controllable network device comprising a parser and a service provider, said parser including instructions for selectively invoking said service provider in response to a command parsed from an external input received by said network device.

2. The network device of claim 1, wherein the external input comprises an HTTP-compliant message.

3. The network device of claim 1, wherein said network device is arranged between a client and a server, said parser further including instructions for selectively invoking said service provider in response to a command parsed from a request received from the client.

4. The network device of claim 1, wherein said network device is arranged between a client and a server, said parser further including instructions for selectively invoking said service provider in response to a command parsed from a data object received from the server.

5. The network device of claim 4, wherein said service provider comprises instructions for transcoding said data object prior to forwarding said data object to the client.

6. The network device of claim 1, wherein said network device is arranged between a client and a second network device, said parser further including instructions for selectively invoking said service provider in response to a command parsed from a data object received by said network device from the second network device.

7. The network device of claim 1, wherein said service provider comprises instructions for reconfiguring said network device.

8. The network device of claim 7, wherein said service provider further comprises instructions for setting an operational parameter of said network device.

9. The network device of claim 1, wherein said service provider further comprises instructions for managing a log file used by said network device.

10. The network device of claim 1, wherein said service provider further comprises instructions for obtaining status information from said network device.

11. The network device of claim 1, wherein said service provider further comprises instructions for restarting said network device.

12. The network device of claim 1, wherein said network device further comprises a network proxy.

13. The network device of claim 1, wherein said network device further comprises a content server.

14. The network device of claim 1, wherein said network device further comprises a client device.

15. A method for dynamically controlling a network device, wherein the network device is coupled to a second network device, said method comprising the steps of:

receiving an input from the second network device;

extracting a command from said input; and

performing an action in response to said extracted command.

16. The method of claim 15, wherein said step of receiving an input comprises receiving an HTTP-compliant message.

17. The method of claim 15, wherein said step of performing an action comprises reconfiguring the network device.

18. The method of claim 17, wherein said step of receiving an input comprises receiving a result of an HTML form.

19. The method of claim 15, wherein said step of performing an action comprises transcoding a data object and transmitting said transcoded data object to another device.

20. A storage medium including a set of instructions for execution by a network device, wherein the network device is coupled to a second network device, said set of instructions comprising instructions for:

receiving an input from the second network device;

extracting a command from said input; and

performing an action in response to said extracted command.

21. The storage medium of claim 20, wherein the storage medium comprises a magnetic storage device.

22. The storage medium of claim 20, wherein the storage medium comprises a memory coupled to the computer.

23. The storage medium of claim 20, wherein the storage medium comprises an optical storage media.

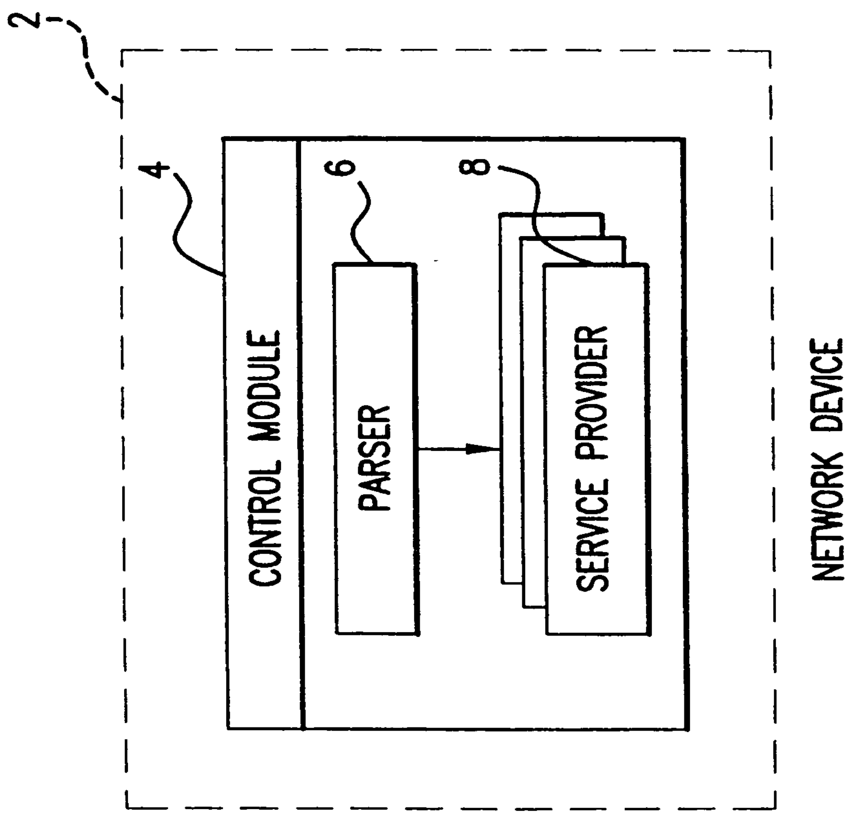

2
4
6
8
CONTROL MODULE
PARSER
SERVICE PROVIDER
NETWORK DEVICE

FIG.1

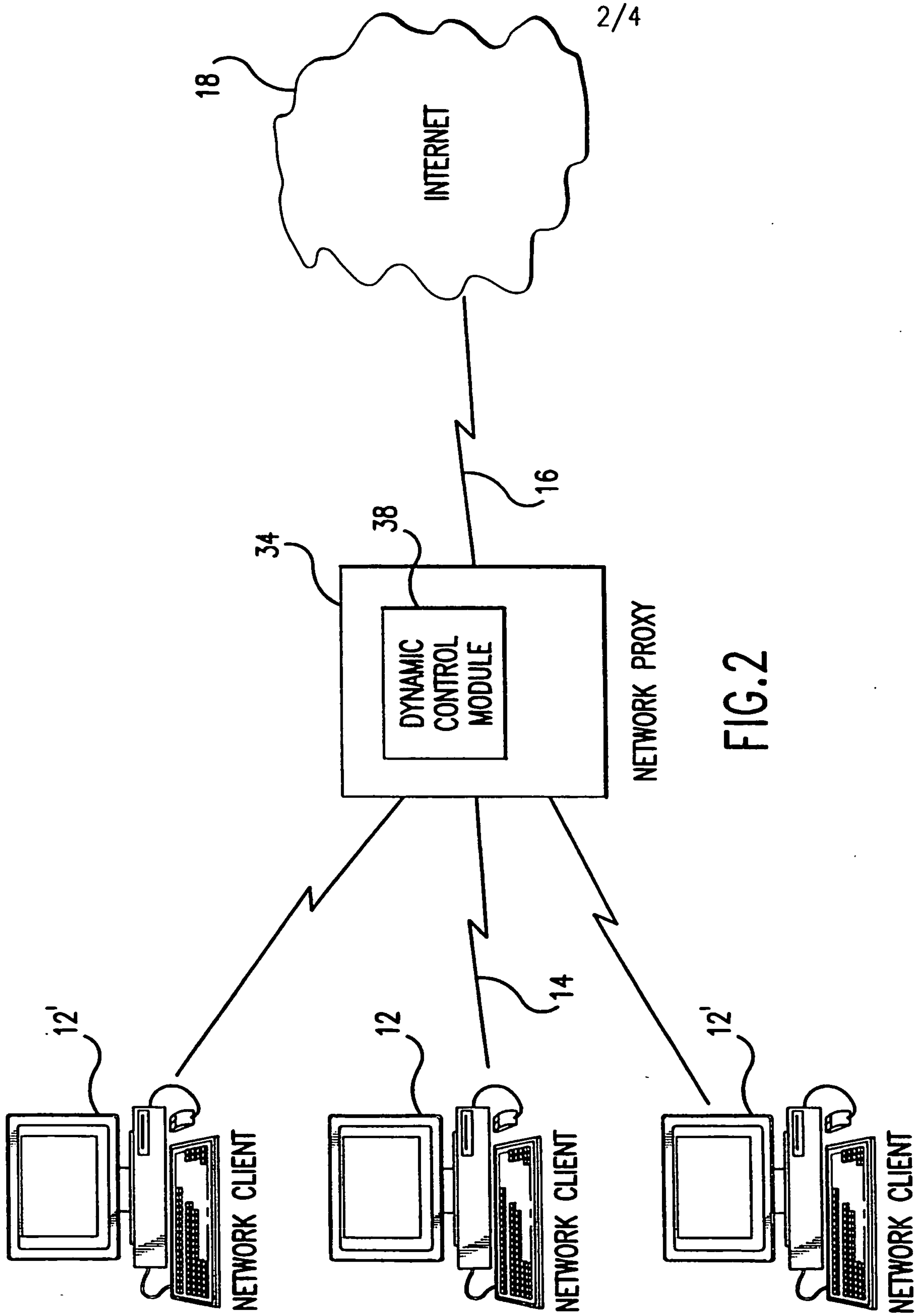

INTERNET
18
16
34
38
DYNAMIC
CONTROL
MODULE
NETWORK PROXY
12'
12
14
12'
NETWORK CLIENT
NETWORK CLIENT
NETWORK CLIENT

FIG.2

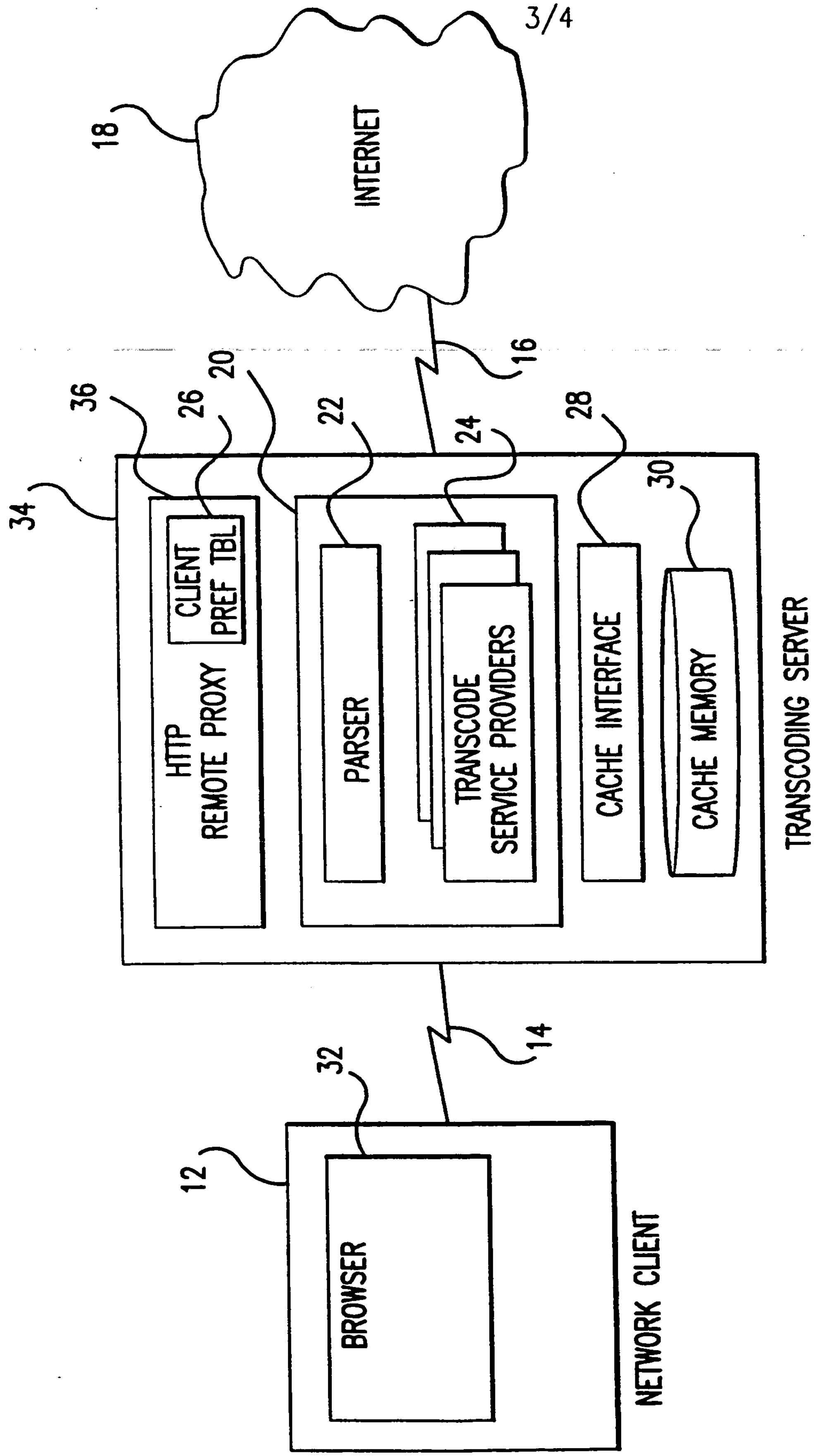

INTERNET
18
16
20
36
26
22
24
28
30
34
HTTP
REMOTE PROXY
CLIENT
PREF TBL
PARSER
TRANSCODE
SERVICE PROVIDERS
CACHE INTERFACE
CACHE MEMORY
TRANSCODING SERVER
14
32
12
BROWSER
NETWORK CLIENT

FIG.3

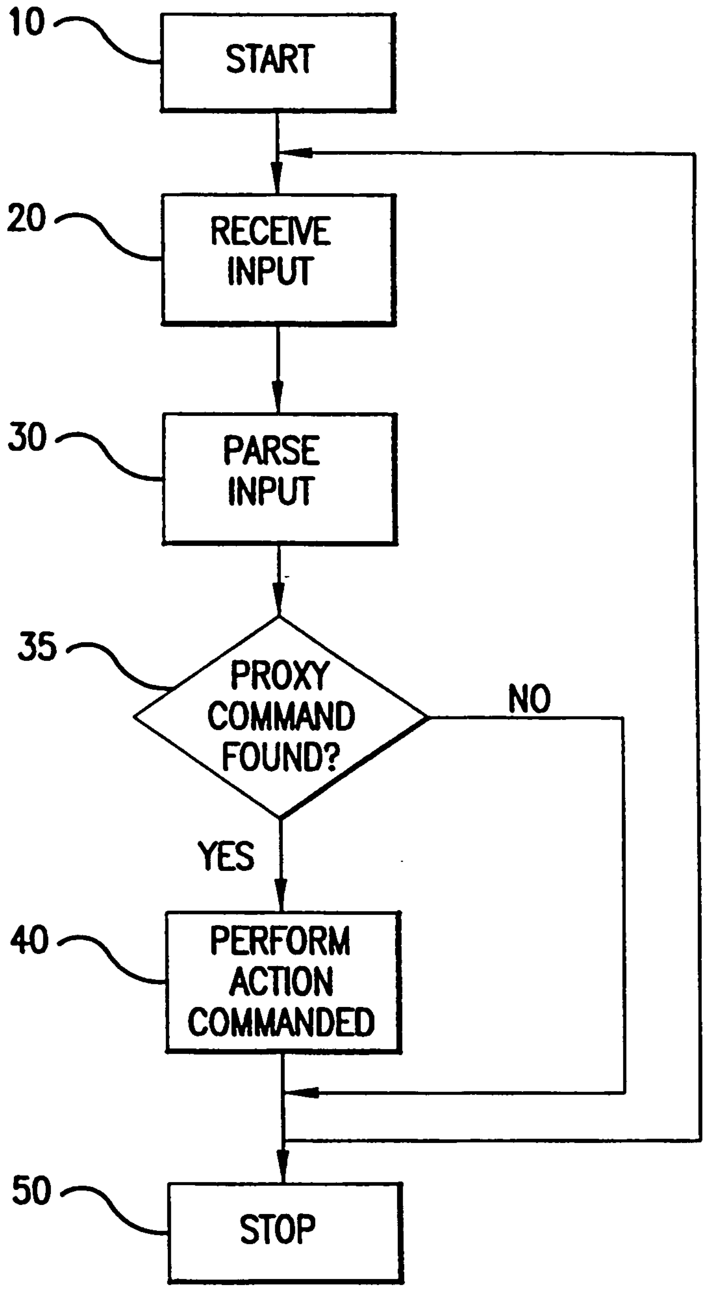

10
START
20
RECEIVE INPUT
30
PARSE INPUT
35
PROXY COMMAND FOUND?
NO
YES
40
PERFORM ACTION COMMANDED
50
STOP

FIG.4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/05456

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 13/38, 15/17
US CL :395/200.53, 200.48; 345/335
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/ 200.33, 200.47, 200.48, 200.54; 345/335

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

NONE

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

USPTO APS
search terms: WWW, URL, HTML, HTTP, parser, configure, data object, selection criteria, forward

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,544,320 A (KONRAD) 6 August 1996 Abstract; Fig. 2; C16:14-C17:18	1-23
Y, E	US 5,706,434 A (KREMEN et al.) 6 January 1998 Abstract; Fig. 1, 5, 6; C5:21-59; C10:23-C11:50	1-23
A, P	US 5,724,556 A (SOUDER et al.) 3 March 1998 Abstract; Fig. 6, 9	1-23
X, E	US 5,768,510 A (GISH) 16 June 1998 Abstract; Fig. 6-7, 11-13, 26; C5:24-50; C18:60-C20:18; C22:49-C23:38; C25:66-C26:20	1-23

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
03 JULY 1998	02 SEP 1998

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	MARK H. RINEHART
Facsimile No. (703) 305-3230	Telephone No. (703) 305-9700